

# Force-based Cooperative Search Directions in Evolutionary Multi-objective Optimization<sup>\*</sup>

Bilel Derbel<sup>1,2</sup>    Dimo Brockhoff<sup>1</sup>    Arnaud Liefooghe<sup>1,2</sup>

<sup>1</sup> Inria Lille - Nord Europe, DOLPHIN project-team, 59650 Villeneuve d'Ascq, France

<sup>2</sup> Université Lille 1, LIFL, UMR CNRS 8022, 59655 Villeneuve d'Ascq cedex, France  
firstname.lastname@inria.fr

**Abstract.** In order to approximate the set of Pareto optimal solutions, several evolutionary multi-objective optimization (EMO) algorithms transfer the multi-objective problem into several independent single-objective ones by means of scalarizing functions. The choice of the scalarizing functions' underlying search directions, however, is typically problem-dependent and therefore difficult if no information about the problem characteristics are known before the search process. The goal of this paper is to present new ideas of how these search directions can be computed *adaptively* during the search process in a *cooperative* manner. Based on the idea of Newton's law of universal gravitation, solutions attract and repel each other *in the objective space*. Several force-based EMO algorithms are proposed and compared experimentally on general bi-objective  $\rho$ MNK landscapes with different objective correlations. It turns out that the new approach is easy to implement, fast, and competitive with respect to a  $(\mu + \lambda)$ -SMS-EMOA variant, in particular if the objectives show strong positive or negative correlations.

## 1 Introduction

Besides established *Pareto-based* EMO algorithms, such as NSGA-II, SPEA2, or  $\varepsilon$ -MOEA, and the recently proposed *indicator-based* algorithms such as IBEA, SMS-EMOA, MO-CMA-ES, or HypE, a third group of *aggregation-based* algorithms, containing e.g. MSOPS [5] and MOEA/D [17], for solving multi-objective optimization problems can be identified [14]. Aggregation-based algorithms reformulate the multi-objective optimization problem as a set of single-objective problems by means of multiple scalarizing functions [10] that are typically solved independently from each other. Standard scalarizing functions such as weighted sum or achievement functions are thereby defining a search direction in the objective space in which the solutions evolve during the search process. As the a priori definition of these search directions is difficult if no further information is known about the problem at hand, this paper proposes a new force-based approach to cooperatively adapt the (single-objective) search directions.

Throughout the paper, we assume the maximization of a vector-valued objective function  $\phi : X \rightarrow Z$  that maps a solution  $x$  from the feasible *search space*  $X$  to  $\phi(x) = (\phi_1(x), \dots, \phi_M(x))$  in the *objective space*  $Z \subseteq \mathbb{R}^M$ . We say, an objective

---

<sup>\*</sup> This is an author version of the EMO'2013 paper published by Springer Verlag. The final publication is available at [www.springerlink.com](http://www.springerlink.com).

vector  $z \in Z$  is *dominated* by objective vector  $z' \in Z$ , denoted by  $z \prec z'$ , if for all  $i \in \{1, \dots, M\}$   $z_i \leq z'_i$  and there exists a  $j \in \{1, \dots, M\}$  such that  $z_j < z'_j$ . Similarly, a solution  $x \in X$  is dominated by  $x' \in X$ , denoted by  $x \prec x'$ , if  $\phi(x) \prec \phi(x')$ . An objective vector  $z \in Z$  is *non-dominated* with respect to a set  $S$  if there does not exist any other objective vector  $z' \in S$  such that  $z \prec z'$ . A solution  $x \in X$  is *Pareto optimal* if  $\phi(x)$  is non-dominated with respect to  $Z$ . The set of all Pareto optimal solutions is the *Pareto set* and its mapping in the objective space is the *Pareto front* of which an approximation is sought.

**Contribution Overview.** Generally speaking, we consider to evolve a set of solutions towards the Pareto front by computing, dynamically at each generation, a force-based direction in the objective space with respect to each solution. Each direction is used to define a single-objective optimization problem to be optimized by each solution independently of the others. Inspired by particle physics and more precisely by Newton laws, we define the direction relative to a solution as an aggregation of forces exerted by other solutions in the objective space. For each pair of solutions, we propose to compute their respective forces according to their dominance relation, while adjusting force magnitudes according to the distance between solutions in the objective space. One specificity of this approach is to evolve a set of solutions in a dynamic and local manner. In fact, search directions are not fixed and evolve throughout generations in an attempt to adaptively fit the search process and better approach the Pareto front. Furthermore, while search directions are computed in a cooperative manner depending on the relative position of solutions at some point of the search, each solution uses its own direction in parallel to other solutions in order to evolve towards a new solution. Thus, maintaining the set of solutions is done in a straightforward manner while avoiding sophisticated data structures and costly operations. Besides being extremely simple to implement, and through extensive experiments on  $\rho$ MNK landscapes, our approach is also proved to be efficient in dynamically finding good directions leading to a good approximation of the Pareto front.

**Related Work.** While the force-based approach presented in this paper share similarities with particle swarm optimization, there are few other studies that are even more related and also compute forces among solutions to steer the search and maintain diversity. In [15, 16], for example, the authors present a constraint multi-objective artificial physics optimization algorithm for continuous problems extending on previous single-objective techniques based on virtual force computations. Such an algorithm considers to move each individual using a velocity vector driven by the total force from other individuals *in the search space*. Similar ideas are investigated in the so-called gravitational search algorithm [4, 11, 12]. One can also find other related studies where forces are used in a *problem-specific* manner in order to move solutions in the search space [3, 8]. However, the way forces are computed and used by those algorithms differ mainly in the fact that the approach proposed here is *problem-independent*. Indeed, it computes the forces and search directions *in the objective space* and then evolves individuals *adaptively* in a *cooperative manner* on the basis of those computed directions.

**Outline.** The paper is organized as follows. Section 2 proposes the general template of the force-based EMO algorithm together with different instantiations of its compo-

---

**Algorithm 1:** Force-Based EMO generic scheme
 

---

```

1  $P \leftarrow \{x^1, x^2, \dots, x^\mu\}$ : initial population;
2  $Z \leftarrow \phi(P) = \{z^1, z^2, \dots, z^\mu\}$ : initial outcome vector in the objective space;
3 repeat
4   for  $i \in \{1, \dots, \mu\}$  do
5      $\vec{d}^i \leftarrow \text{Force}(z^i, Z \setminus \{z^i\})$ ;
6   for  $i \in \{1, \dots, \mu\}$  do
7      $S^i \leftarrow \text{Pool}(x^i) \cup \{x^i\}$  /* variation */
8      $x^i \leftarrow \text{Select}(S^i, \vec{d}^i)$  /* replacement selection */
9 until STOPPING_CONDITION;
```

---

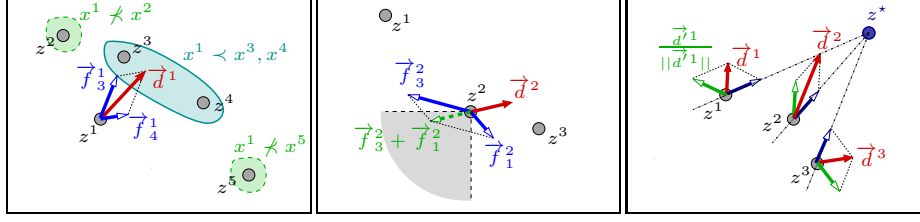
nents. In particular, a number of force-based search direction schemes and different selection and replacement strategies are introduced. Section 3 presents the setup of the experimental analysis. Section 4 discusses the dynamics of the algorithm with the aim of better understanding its general behavior. Section 5 gives a detailed experimental analysis. Finally, Section 6 concludes the paper and discusses further research.

## 2 Algorithm Description

The basic idea behind our approach is to view the evolution of individuals from one generation to another as a set of particles moving *in the objective space* due to virtual forces exerted by other particles in the population. Algorithm 1 gives a high-level description of such a force-based EMO algorithm. Starting with an initial population, the algorithm proceeds in generations in which the population is evolved by means of three main steps. First, for each individual  $x^i$ , we compute a search direction  $\vec{d}^i$  using function **Force** and the position of other particles in the objective space. Then, for each individual  $x^i$ , a sample of candidate solutions  $S^i$  is, independently of the computed search directions, generated by means of some variation operator(s), denoted by **Pool**. Finally, the selection of the new candidate solution from the old individual  $x^i$  and its offsprings  $S^i$ , denoted by **Select**, is based on an underlying scalar sub-problem defined by the force-based direction  $\vec{d}^i$  previously computed for  $x^i$ . In the following, we give a detailed description of the functions **Force** and **Select**. Function **Pool** is independent of the direction and typically problem-specific.

### 2.1 Force-based Search Direction Strategies

The way we define the search directions  $\vec{d}^i, i \in \{1, \dots, \mu\}$ , is crucial to control the movement of particles, and to efficiently guide them towards the Pareto front. Compared against a simple strategy where search directions are fixed initially, for instance following a random distribution, we consider to dynamically and adaptively compute search directions following some attraction-repulsion force-based rules. Roughly speaking, the closer particles corresponding to non-comparable individuals are in the objective space,



**Fig. 1.** Illustration of force strategies: D-D (left), NB-D (middle) and BH-D (right).

the more particles should move away from each others to increase diversity—meaning that search directions of individuals should, in general, be relatively repulsive. On the other hand, particles should also move towards better objective function values and search directions should be attractive with respect to dominating individuals. In this study, we use different **Force** functions in order to compute directions which shall induce different dynamics of the particles, and thus provide different results in terms of approximation quality. In the following, we assume that we are given a norm function  $\|\cdot\|$  in the objective space and a scaling factor  $\alpha \in \mathbb{R}^+$ . We denote the force exerted by particle  $z^j$  (w.r.t. solution  $x^j$ ) on particle  $z^i$  by  $\vec{f}_j^i$  and define the five following strategies.

**Repulsive force-based directions (R-D).** Within R-D forces, particles are pairwise repulsive. More precisely, for every  $i, j \in \{1, \dots, \mu\}$ , we set:

$$\vec{f}_j^i = \frac{z^i - z^j}{\|z^i - z^j\|^\alpha} . \quad (1) \quad \vec{d}^i = \sum_{j \in \{1, \dots, n\} \setminus \{i\}} \vec{f}_j^i . \quad (2)$$

**Repulsive-attractive directions (RA-D).** Here, a particle is attracted by any other particle which is dominating it and repelled otherwise. More precisely, for every  $i \in \{1, \dots, \mu\}$ , the direction  $\vec{d}^i$  is given by Eq. 2 and the forces  $\vec{f}_j^i$  are defined as:

$$\vec{f}_j^i = \begin{cases} \frac{z^j - z^i}{\|z^i - z^j\|^\alpha} & \text{if } x^i \prec x^j \\ \frac{z^i - z^j}{\|z^i - z^j\|^\alpha} & \text{otherwise} \end{cases} . \quad (3)$$

**Dominance-based directions (D-D).** In the case of the D-D **Force** function, a particle, that is dominated by at least one other particle, is attracted only by those particles dominating it and neither attracted nor repelled by the others. Otherwise, if a particle is not dominated by any other particle, then its direction is computed following the previous repulsive-attractive (RA-D) strategy. This strategy is motivated by the intuitive idea that particles should exclusively follow those that are dominating them, in an attempt to intensify the search process (see Fig. 1). More formally, let  $\text{Dom}(x^i)$  be true whenever there exists a  $k \neq i$  such that  $x^i \prec x^k$ , and false otherwise. Direction  $\vec{d}^i$  for every

particle is then given by Eq. 2 where forces  $\vec{f}_j^i$  are now defined as follows:

$$\vec{f}_j^i = \begin{cases} \frac{z^j - z^i}{\|z^i - z^j\|^\alpha} & \text{if } \text{Dom}(x^i) \text{ and } x^i \prec x^j \\ 0 & \text{if } \text{Dom}(x^i) \text{ and } \neg(x^i \prec x^j) \\ \frac{z^i - z^j}{\|z^i - z^j\|^\alpha} & \text{otherwise} \end{cases} \quad (4)$$

**Non-backward directions (NB-D).** The repulsive-attractive force-based rule of Eq. 3 can lead to situations where a direction  $\vec{d}^i$  computed for a particle  $z^i$ , which is *not dominated* by other particles, points away from the Pareto front. In the case of Fig. 1, for example, the direction computed for  $z^2$  may lead to a situation where it is replaced by a dominated particle during the search. To counteract this situation, the NB-D strategy inverts those directions which are going backward and uses the direction  $\vec{d}^i = -\vec{1} \cdot \vec{d}^i$  if  $\vec{d}^i \cdot z^i \prec z^i$  and  $\vec{d}^i = \vec{d}^i$  otherwise, instead of the standard repulsive-attractive directions  $\vec{d}^i$  computed with Eq. 2 and 3.

**Black hole directions (BH-D).** Here, we consider an imaginary fixed particle which acts as a black hole attracting all the others. More precisely, we consider a virtual particle at the position of a utopian point  $z^*$  such that for every  $i$ ,  $z^i \prec z^*$  [10] which additionally attracts all other particles when computing search directions. With the directions  $\vec{d}^i$  computed via Eq. 3 and Eq. 2, the BH-D strategy then uses the directions:

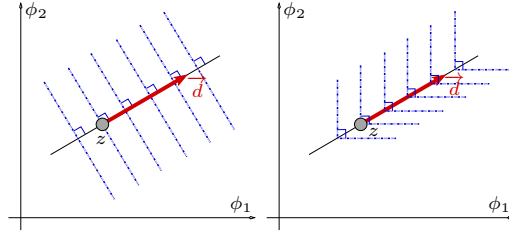
$$\vec{d}^i = \vec{d}^i / \|\vec{d}^i\| + (z^* - z^i) / \|z^* - z^i\| \quad (5)$$

## 2.2 Selection and Replacement Strategies

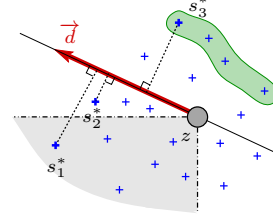
Once the direction of each particle is computed, Algorithm 1 proceeds to the computation of the next generation. Each solution  $x^i$  is replaced by a new one from set  $S^i$ , using its direction  $\vec{d}^i$  and function **Select**. We shall consider several strategies, which are essentially a mix between two ideas. Firstly, we shall use one of two scalarizing (single-objective) functions denoted by  $\mathcal{W}$  and  $\mathcal{A}$  to evaluate the candidate solutions. These two functions are based on a weighted sum and an achievement scalarizing function [10], where the weighting coefficient vector used for each solution  $x^i$  is determined by the corresponding direction  $\vec{d}^i = \{d_1^i, \dots, d_M^i\}$  of particle  $z^i = \{z_1^i, \dots, z_M^i\}$ . More formally, given a candidate solution  $x \in S^i$  w.r.t. individual  $x^i$ , we let:

$$\mathcal{W}(x) = \sum_{m=1}^M d_m^i \cdot \phi_m(x) \quad \text{and} \quad \mathcal{A}(x) = \max_{m \in \{1, \dots, M\}} \{w_m^i \cdot (z_m^i - \phi_m(x))\} \quad (6)$$

where  $w_m^i = 1/d_m^i$  if  $d_m^i \neq 0$  and  $w_m^i = 0$  otherwise. Notice that with function  $\mathcal{W}$ , the lines of equal fitness values are orthogonal to the search direction  $\vec{d}^i$ , no matter the sign of  $d^i$  components. With  $\mathcal{A}$ , they are half-lines reaching out from the line through  $z^i$  in direction  $\vec{d}^i$  (see Fig. 2).



**Fig. 2.** Lines of equal fitness values for functions  $\mathcal{W}$  (left), and  $\mathcal{A}$  (right).



**Fig. 3.** Selection strategies (i), (ii), and (iii) based on a weighted sum scalarizing function with the selected objective vectors  $s_1^*$  for strategy (i),  $s_2^*$  for (ii), and  $s_3^*$  for (iii).

Secondly, we shall choose from a subset  $S'^i \subseteq S^i$  of candidate solutions, the solution that optimizes the so-defined scalar problem. We here consider three different possibilities with an increasing inherent focus towards the Pareto front, (see also Fig. 3 for an example): (i)  $S'^i = S^i$ , (ii)  $S'^i = \{x \in S^i \mid x \not\prec x^i\}$  the subset of candidate solutions not dominated by  $x^i$ , or (iii)  $S'^i = \{x \in S^i \mid \forall x' \in S^i \setminus \{x\}, x \not\prec x'\}$  the set of candidate solutions from  $S^i$  which are not dominated. The output of function **Select** can then be formalized using the simple single-objective problem:

$$\text{Select} \left( S^i, \vec{d}^i \right) = \operatorname{argopt}_{x \in S'^i} G(x) \quad (7)$$

where  $G \in \{\mathcal{W}, \mathcal{A}\}$  and  $S'^i$  is one of the previously defined subsets. Notice that overall we have six possible combinations leading to six variants of function **Select**.

### 3 Experimental Design

In order to better understand and compare the different force, selection, and replacement strategies, we follow two separate lines of presentation. In Section 4 we show some exemplary runs in detail while Section 5 presents quality performance from 30 independent runs and the results of statistical tests. With this section, we provide details on the experimental design of those comparisons with respect to the used algorithms and test problem instances, the parameter setting, and performance assessment.

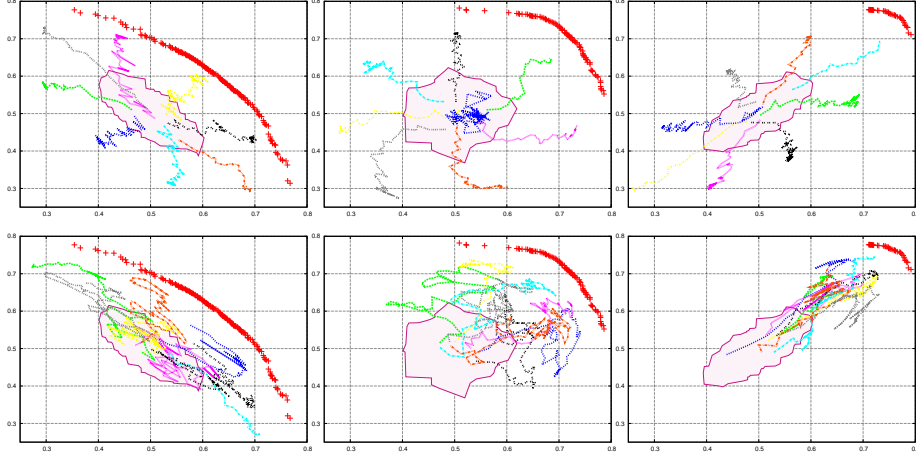
**Competing Algorithms.** Besides the five force variants R-D, RA-D, NB-D, D-D, and BH-D with their two different scalarizing functions (weighted sum, achievement) and the three proposed selection and replacement strategies, we use two baseline algorithms in our comparisons. On the one hand, we have a simple strategy, denoted by I-D, which assigns a fixed search direction to each of the  $\mu$  population members. The search directions are *initially fixed* to a set of  $\mu$  direction vectors chosen with equal distances in the weight space. The same functions **Pool** and **Select** as for the force-based strategies are used for algorithm I-D, except that the weights of the functions  $\mathcal{W}$  and  $\mathcal{A}$  are fixed throughout the search and instead of  $z^i$  in Eq. 6, a utopian point  $z^*$  is used. This corresponds to the classical way of defining multiple independent scalarizing functions and allows us to appreciate the gain we get when adapting search directions.

The second baseline algorithm is a  $(\mu + \lambda)$ -SMS-EMOA [2] variant with one-shot replacement strategy—denoted by SMS. At each iteration of the algorithm,  $\lambda$  solutions are selected (with replacement) and mutated. The  $\mu$  solutions forming the population of the next iteration are chosen from the old  $\mu$  and the new  $\lambda$  solutions by means of their hypervolume contributions after a non-dominated sorting and using the contributing hypervolume as the second-level sorting criterion. All  $\mu$  solutions are chosen at once without new hypervolume calculations (one-shot scenario). Compared to our *local* approach where each solution of the population is replaced independently of the other ones, SMS uses a *global* strategy to evolve the whole population of individuals—thus allowing us to better appreciate the locality property of our approach.

**$\rho$ MNK-landscapes.** The family of  $\rho$ MNK-landscapes is a problem-independent model used for constructing multi-objective multi-modal landscapes with objective correlation [13]. It extends single- and multi-objective NK-landscapes [1, 7]. Feasible solutions are represented as binary strings of size  $N$ , *ie.* the decision space is  $X = \{0, 1\}^N$ . Parameter  $K$  refers to the number of variables that influence a particular position from the bit-string (the epistatic interactions). Each objective function  $\phi_m : \{0, 1\}^N \rightarrow [0, 1]$  is defined by  $\phi_m(x) = \frac{1}{N} \sum_{i=1}^N c_i^m(x_i, x_{i_1}, \dots, x_{i_K})$  where  $c_i^m : \{0, 1\}^{K+1} \rightarrow [0, 1]$  defines the multidimensional component function associated with variable  $i \in \{1, \dots, N\}$ , and where  $K < N$  [13]. By increasing the number of variable interactions  $K$  from 0 to  $(N - 1)$ ,  $\rho$ MNK-landscapes can be gradually tuned from smooth to rugged. In this work, we choose the positions of these interactions uniformly at random and the same for all objective functions. Component values are distributed in the range  $[0, 1)$  and follow a multivariate uniform distribution of dimension  $M$ , defined by a correlation coefficient  $\rho > \frac{-1}{M-1}$ . The positive (resp. negative) data correlation decreases (resp. increases) the degree of conflict between objectives.

**Parameter Setting.** In this paper, we consider three bi-objective ( $M = 2$ )  $\rho$ MNK-landscapes of size  $N = 128$ , non-linearity  $K = 4$ , and correlation  $\rho \in \{-0.7, 0.0, +0.7\}$ . One instance, generated at random, is considered per parameter setting. For all competing algorithms, we consider an independent bit-flip mutation operator, where each bit is mutated at random with a probability  $1/N$ . We use the Euclidean distance and  $\alpha = 2$  to compute forces. The utopia point  $z^*$  is set to  $(1, 1)$ . For R-D, RA-D, NB-D, D-D, BH-D, and I-D, the candidate solution set, *i.e.*, **Pool**, is obtained by mutating each solution  $N$  times. For algorithm SMS,  $\lambda$  is set to  $N$ . All force-based algorithms are run for a fixed number of generations denoted by  $gen \leq 128$ . In order to compare the algorithms in terms of function evaluations fairly, one generation of Algorithm 1 is equivalent to  $\mu$  generations of SMS. For each algorithm run, we use an unbounded archive to record the computed Pareto front approximations. We shall use  $\mu \in \{8, 16, 32, 64, 128, 256, 512\}$  as population sizes. We use the notation  $F \in \{\mathcal{W}, \mathcal{A}\}$  to refer to weighted sum ( $F = 0$ ) and achievement scalarizing function ( $F = 1$ ). Similarly,  $S \in \{0, 1, 2\}$  indicates which selection strategy of Section 2.2 is used ( $S = 0$ : all,  $S = 1$ : solutions not dominated by the parent,  $S = 2$ : solutions not dominated by the parent and the offspring).

**Performance Assessment.** A set of 30 runs *per* instance is performed for each algorithm using a randomly generated initial population. For each  $\rho$ MNK-landscape and a given number of function evaluations, we compute a reference set  $Z_N^*$  containing the



**Fig. 4.** Exemplary trajectories of R-D (top) and RA-D (bottom) strategies. From left to right:  $\rho = -0.7$ ,  $\rho = 0.0$ , and  $\rho = +0.7$ . Polygons are obtained by random sampling for an equivalent number of evaluations. Runs are for  $\mu = 8$ ,  $F = 0$ ,  $S = 0$ ,  $gen = 64$ . Crosses represent the reference set  $Z_N^*$ .

non-dominated points among all solutions visited during all experiments. To measure the quality of a Pareto front approximation  $A$  in comparison to  $Z_N^*$ , we use both the difference hypervolume indicator ( $I_H^-$ ) and the multiplicative epsilon indicator ( $I_\epsilon^\times$ ) [18]. The  $I_H^-$ -indicator gives the portion of the objective space that is dominated by  $Z_N^*$  and not by  $A$ . The reference point is set to the origin. The  $I_\epsilon^\times$ -indicator gives the minimum multiplicative factor by which an approximation  $A$  has to be translated in each dimension of the objective space in order to dominate the reference set  $Z_N^*$  (both  $I_H^-$ - and  $I_\epsilon^\times$ -values are to be minimized). The experimental results report the average indicator value together with the results of pairwise Mann-Whitney tests with a  $p$ -value of 0.05.

To explore the difference between the multiple variants of the competing algorithms, we also consider their empirical attainment functions (EAFs) [9]. The EAF provides the probability, estimated from several runs, that an arbitrary objective vector is dominated by, or equivalent to, a solution obtained by a single run of the algorithm. The difference between the EAFs for two different algorithms enables to identify the regions of the objective space where one algorithm performs better than another. The magnitude of the difference in favor of one algorithm is plotted within a gray-colored graduation.

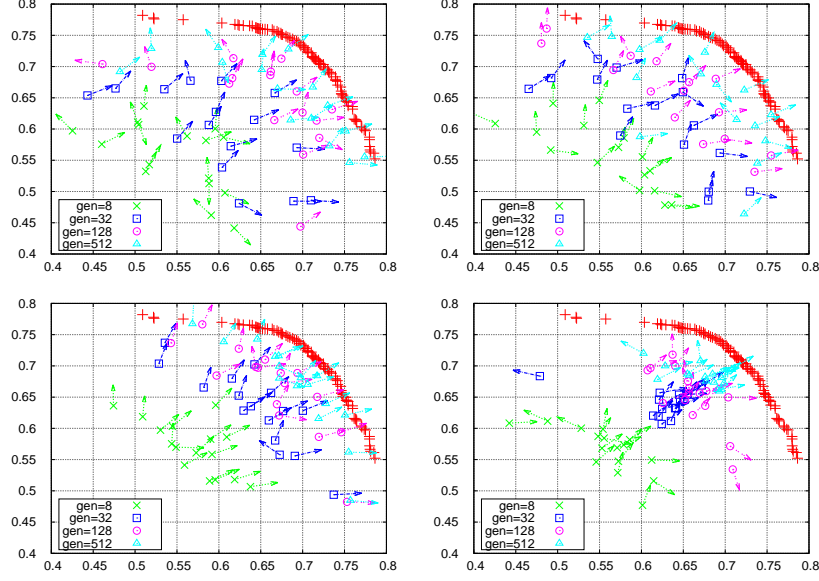
## 4 Understanding the Algorithm

In this section, we report some exemplary runs to better understand the idea of our approach and the role of the different selection strategies and search directions.

### 4.1 Feasible Objective Space vs. Non-dominated Set Approximation

Fig. 4 shows the trajectories of the  $\mu = 8$  population members in exemplary runs of the R-D and RA-D strategies on  $\rho$ MNK landscapes with different correlation  $\rho$  and up to 64





**Fig. 5.** Exemplary directions (unitary arrows) at different generations: RA-D (top left), NB-D (top right), BH-D (bottom left) and D-D (bottom right). Runs are for  $\rho = 0.0$ ,  $\mu = 16$ ,  $F = 0$ ,  $S = 0$ .

generations. The polygons in the background show the spread<sup>3</sup> of  $gen \cdot N \cdot \mu = 65,536$  random samples which correspond to the same amount of function evaluations as for the force-based strategies after  $gen = 64$  generations.

We can make two main observations. First, the attracting forces of the RA-D strategy seem to “boost” the population towards the Pareto front while only using repelling forces in R-D allows to get a good idea about the feasible objective space region which can be considered as of independent interest [6]. Secondly, Fig. 4 shows how much more efficient the force-based algorithms are when compared to naive random sampling. In the exemplary results, the R-D strategy covers an objective space area more than four times larger when compared to random sampling while the RA-D variant roughly halves the distance to the best known Pareto front approximation—independently of the objective correlation  $\rho$ . Note that the plots are conceptionally similar when looking at different runs or different settings of  $\mu$ ,  $F$ , or  $S$ .

## 4.2 Comparing Force-based Search Directions

In Fig. 5, we report exemplary runs showing how solutions and their directions evolve through generations. We can observe that the direction distribution of all strategies is rather stable over time—except for D-D where directions get clustered around the diagonal as soon as some dominating solutions are found. However, between the considered

<sup>3</sup> The polygon corners are defined by the non-dominated solutions found when maximizing (minimizing) both objectives and by the non-dominated solutions found when maximizing one and minimizing the other.

strategies, differences can be noticed with respect to both population distribution and directions. It appears that the strategies RA-D and NB-D cover a larger area of the objective space when compared to BH-D and D-D. The directions produced by RA-D and NB-D also appear to be more diversified. The D-D strategy has a tendency to cluster solutions in the middle, while solutions near the extremes of the reference set  $Z_N^*$  are rare. This is without surprise since the D-D strategy follows the idea that solutions are attracted by the dominating ones. Another interesting observation is that for BH-D and NB-D, the search directions appear to be well correlated with the solution positions—pointing right towards the closest solutions in the reference set. However, compared to RA-D, solutions are more clustered near the center. As it will be shown later, the relative performance of our different force strategies are directly related to the behaviors observed informally in this set of exemplary runs.

## 5 Experimental Analysis

In this section, we go into a deeper and more detailed experimental analysis of our approach by considering its performance over multiple runs. We first start with Table 1, providing the values of indicators  $I_H^-$  and  $I_\epsilon^\times$  for the instance with correlation  $\rho = -0.7$ , and for different algorithms under different configurations.

**Influence of the Neighborhood Selection Strategy.** A very first observation can be made on the minor effect of the neighborhood selection strategy. In fact, only in exceptional cases does the strategy have a strong influence on the indicator values and the algorithm ranking. For instance, this is the case for a low number of generations or for the I-D strategy (fixed independent weights) where the strategy of selecting the non-dominated solutions ( $S = 2$ ) appears to be better than the two others. When examining the results for correlations  $\rho = 0.0$  and  $\rho = 0.7$ , essentially the same observation can be made. Thus, for these correlations we show in Table 2 the results using the most simplest implementation only ( $S = 0$ , i.e., selection among all candidates).

**Weighted Sum vs. Achievement Scalarizing Function.** When comparing the algorithm variants employing weighted sum ( $F = 0$ ) with the ones using achievement function ( $F = 1$ ), we can clearly see that the weighted sum consistently yields better results in terms of hypervolume and  $\epsilon$ -indicator differences (see Tables 1 and 2). To understand this outperformance, let us again consider the lines of equal utility in Fig. 2. For a search direction pointing towards the Pareto front, it can be observed that the weighted sum allows a particle to move to incomparable solutions while with the achievement function, a particle can only move to a dominating solution. Thus, assuming that for the  $\rho$ MNK landscapes, the bit-flip operator is likely to produce more incomparable neighbors than dominating ones, we can reasonably claim that it is more difficult for our algorithms to escape local optima using the achievement scalarizing function.

**Comparison between the Five Scalarizing Strategies.** One observation is that the algorithm based on  $N$  independently fixed scalarizing functions results in the worst hypervolume and  $\epsilon$ -indicator values—except for low and medium numbers of function evaluations if only the non-dominated portion of the offspring is considered for

**Table 1.** Comparison of the different algorithms with respect to indicators  $I_H^-$  and  $I_\epsilon^\times$  (lower is better) for the anti-correlated instance ( $\rho = -0.7$ ),  $\mu = 128$  and  $\text{gen} \in \{8, 16, 32, 64, 128\}$ . Column  $r$  is the rank of the algorithm under consideration—computed as the number of algorithms that statistically outperform it with the given indicator. Bold style refers to algorithms that are not outperformed by any other. Gray cells refer to algorithms which are analyzed in Fig. 8 and Fig. 9.

		gen=8				gen=16				gen=32				gen=64				gen=128					
		F	S	$I_{\epsilon}^{\times}$	r	$I_H^{-}$	r	$I_{\epsilon}^{\times}$	r	$I_H^{-}$	r	$I_{\epsilon}^{\times}$	s	$I_H^{-}$	r	$I_{\epsilon}^{\times}$	r	$I_H^{-}$	r	$I_{\epsilon}^{\times}$	r	$I_H^{-}$	r
RA-D	0	0	1	1.211	5	0.148	4	1.118	2	0.073	6	1.082	2	<b>0.050</b>	0	<b>1.057</b>	0	<b>0.036</b>	0	<b>1.043</b>	0	<b>0.028</b>	0
RA-D	0	1	1	1.207	2	0.146	2	1.118	2	0.072	2	1.083	4	<b>0.051</b>	0	1.059	2	<b>0.038</b>	0	<b>1.044</b>	0	<b>0.028</b>	0
RA-D	0	2	1	1.207	3	0.145	2	1.119	2	0.072	3	<b>1.079</b>	0	<b>0.051</b>	0	<b>1.057</b>	0	<b>0.038</b>	0	<b>1.044</b>	0	<b>0.029</b>	0
RA-D	1	0	1	1.224	13	0.156	14	1.135	13	0.084	13	1.100	12	0.067	13	1.083	13	0.055	12	1.070	12	0.049	12
RA-D	1	1	1	1.222	11	0.154	13	1.135	13	0.085	13	1.100	12	0.067	14	1.083	13	0.057	15	1.070	12	0.050	13
RA-D	1	2	1	1.221	11	0.155	13	1.135	12	0.085	13	1.099	12	0.069	17	1.083	13	0.059	18	1.069	11	0.051	13
BH-D	0	0	1	1.215	6	0.147	4	1.121	2	0.071	2	<b>1.080</b>	0	<b>0.050</b>	0	1.062	6	0.039	1	1.049	6	0.031	2
BH-D	0	1	1	1.208	3	0.145	2	1.116	2	0.069	1	<b>1.077</b>	0	<b>0.050</b>	0	<b>1.057</b>	0	<b>0.037</b>	0	1.049	3	0.031	2
BH-D	0	2	1	1.211	4	0.147	4	1.126	9	0.073	6	<b>1.082</b>	0	0.052	1	<b>1.058</b>	0	0.039	1	1.051	6	0.032	6
BH-D	1	0	1	1.290	24	0.181	24	1.192	24	0.118	24	1.166	24	0.109	24	1.155	24	0.104	24	1.146	26	0.096	24
BH-D	1	1	1	1.284	24	0.179	24	1.193	24	0.117	24	1.164	24	0.107	24	1.151	24	0.102	24	1.146	26	0.097	24
BH-D	1	2	1	1.290	24	0.181	24	1.201	24	0.121	25	1.167	24	0.109	24	1.150	24	0.102	24	1.146	26	0.096	24
D-D	0	0	1	1.223	11	0.153	13	1.136	12	0.083	13	1.102	12	0.066	13	1.080	11	0.054	11	1.070	11	0.046	10
D-D	0	1	1	1.229	15	0.155	13	1.133	12	0.084	13	1.095	11	0.065	13	1.074	10	0.052	10	1.059	10	0.043	10
D-D	0	2	1	1.215	5	0.149	7	1.125	4	0.076	9	1.086	5	0.058	11	1.065	8	0.046	9	1.053	7	0.037	9
D-D	1	0	1	1.230	16	0.158	17	1.143	20	0.087	18	1.105	18	0.069	14	1.088	19	0.060	20	1.080	19	0.055	20
D-D	1	1	1	1.235	14	0.159	18	1.149	22	0.090	21	1.103	17	0.070	17	1.090	19	0.061	20	1.079	19	0.055	20
D-D	1	2	1	1.234	13	0.156	13	1.142	15	0.089	19	1.109	18	0.073	23	1.089	18	0.061	20	1.077	19	0.053	18
NB-D	0	0	1	1.205	2	0.145	3	1.119	2	0.071	2	1.083	4	0.052	2	<b>1.057</b>	0	<b>0.037</b>	0	<b>1.043</b>	0	<b>0.027</b>	0
NB-D	0	1	1	1.209	3	0.146	3	1.120	2	0.071	2	<b>1.079</b>	0	<b>0.049</b>	0	<b>1.057</b>	0	<b>0.036</b>	0	<b>1.045</b>	0	0.029	2
NB-D	0	2	1	1.203	2	0.144	2	1.115	1	0.071	2	<b>1.079</b>	0	<b>0.051</b>	0	<b>1.056</b>	0	<b>0.038</b>	0	<b>1.043</b>	0	<b>0.029</b>	0
NB-D	1	0	1	1.220	11	0.154	13	1.132	11	0.083	13	1.099	12	0.066	13	1.082	13	0.056	12	1.071	12	0.050	13
NB-D	1	1	1	1.222	12	0.156	13	1.134	13	0.084	13	1.099	12	0.066	13	1.083	13	0.056	12	1.070	12	0.050	13
NB-D	1	2	1	1.227	13	0.155	13	1.134	13	0.085	14	1.100	12	0.070	18	1.082	13	0.057	15	1.070	12	0.052	13
I-D	0	0	1	1.350	27	0.212	27	1.288	28	0.166	27	1.235	27	0.156	27	1.178	27	0.129	27	1.128	23	0.097	24
I-D	0	1	1	1.350	27	0.211	27	1.281	27	0.165	27	1.240	29	0.155	27	1.180	27	0.129	27	1.126	23	0.097	24
I-D	0	2	1	1.200	1	0.140	1	1.111	1	0.067	1	1.085	6	0.053	7	1.075	10	0.050	10	1.067	11	0.047	11
I-D	1	0	1	1.348	27	0.212	27	1.276	27	0.165	27	1.229	27	0.154	27	1.178	27	0.133	29	1.149	26	0.106	29
I-D	1	1	1	1.349	27	0.212	27	1.275	27	0.165	27	1.229	27	0.154	27	1.176	27	0.133	29	1.150	26	0.106	29
I-D	1	2	1	1.193	1	0.143	2	1.129	10	0.071	2	1.105	18	0.057	11	1.095	22	0.052	11	1.085	22	0.049	13
SMS			<b>1.151</b>	0	<b>0.070</b>	0	<b>1.063</b>	0	<b>0.035</b>	0	1.103	12	<b>0.048</b>	0	1.120	23	0.053	11	1.130	23	0.053	18	

selection. This indicates that choosing the search directions cooperatively seems to be a good choice. Secondly, we can observe that, with a few exceptions, in particular for  $\rho = 0.7$  and larger generations, the D-D variant performs badly when compared to the other three force-based algorithms. In these exceptional cases where the D-D variant produces good results for one indicator, the other indicator often shows a medium performance while typically a high positive correlation between the two indicators can be reported. When looking more carefully at how this algorithm approaches the Pareto front (Fig. 5), we can better understand why often the hypervolume values are quite good but the  $\epsilon$ -indicator values are not. In fact, due to the absence of backwards directions and the attraction of dominated by dominating points, the D-D variant loses diversity during the run and approaches the Pareto front via the diagonal of the objective space. Like that, the large number of points in the middle of the objective space results in quite high hypervolume values. However, the absence of extreme solutions in the population yields lower  $\epsilon$ -indicator values.

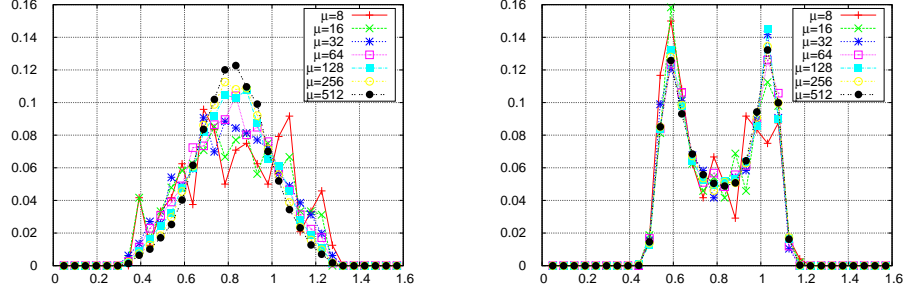
**Table 2.** Comparison of the different algorithms for instance correlation  $\rho \in \{0.0, 0.7\}$ . Same settings and notations than in Table 1 are used. The selection strategy is  $\mathbf{S} = 0$ .

$\rho$		F	gen=8				gen=16				gen=32				gen=64				gen=128			
			$I_{\epsilon}^{\times}$	r	$I_H^{-}$	r	$I_{\epsilon}^{\times}$	r	$I_H^{-}$	r	$I_{\epsilon}^{\times}$	r	$I_H^{-}$	r	$I_{\epsilon}^{\times}$	r	$I_H^{-}$	r	$I_{\epsilon}^{\times}$	r	$I_H^{-}$	r
0.0	RA-D	0	1.215	5	0.176	3	1.138	7	0.109	9	1.085	6	0.063	9	1.058	9	0.038	1	1.041	6	0.027	4
	RA-D	1	1.236	17	0.193	16	1.156	15	0.128	14	1.106	16	0.080	14	1.080	14	0.056	14	1.069	13	0.045	12
	BH-D	0	1.207	2	0.174	3	1.123	1	0.105	3	1.072	1	0.057	1	1.047	1	0.037	1	1.041	6	0.030	7
	BH-D	1	1.227	11	0.189	14	1.149	12	0.132	18	1.114	20	0.097	24	1.101	23	0.084	24	1.093	23	0.077	24
	D-D	0	1.225	11	0.185	13	1.149	12	0.128	14	1.095	13	0.086	18	1.071	12	0.064	21	1.058	12	0.053	20
	D-D	1	1.247	22	0.197	21	1.163	21	0.135	19	1.113	20	0.084	18	1.093	20	0.060	18	1.087	20	0.052	19
	NB-D	0	1.215	6	0.177	3	1.136	5	0.109	7	1.086	7	0.063	9	1.057	9	0.039	5	1.043	7	0.028	4
	NB-D	1	1.235	17	0.192	15	1.154	14	0.128	14	1.102	14	0.079	14	1.081	15	0.055	13	1.070	13	0.045	12
	I-D	0	1.339	27	0.252	27	1.289	27	0.221	27	1.238	27	0.191	27	1.186	27	0.156	27	1.136	27	0.117	27
	I-D	1	1.338	27	0.253	27	1.292	27	0.222	27	1.256	29	0.193	27	1.223	29	0.158	27	1.177	29	0.120	27
	SMS		<b>1.051</b>	0	<b>0.043</b>	0	<b>1.034</b>	0	<b>0.027</b>	0	<b>1.031</b>	0	<b>0.024</b>	0	<b>1.033</b>	0	<b>0.027</b>	0	<b>1.032</b>	0	<b>0.026</b>	0
0.7	RA-D	0	1.208	3	0.180	3	1.118	3	0.103	3	1.060	4	0.048	1	<b>1.040</b>	0	<b>0.035</b>	0	<b>1.032</b>	0	<b>0.024</b>	0
	RA-D	1	1.226	22	0.191	20	1.133	21	0.119	20	1.078	21	0.071	18	1.060	14	0.056	16	1.052	13	0.046	14
	BH-D	0	1.208	3	0.180	3	1.112	2	0.101	3	1.055	1	0.045	1	<b>1.038</b>	0	<b>0.036</b>	0	1.035	3	0.030	6
	BH-D	1	1.212	6	0.184	4	1.119	5	0.108	10	1.068	14	0.062	15	1.060	14	0.058	18	1.055	17	0.053	21
	D-D	0	1.212	5	0.183	4	1.117	3	0.106	5	1.060	3	0.053	9	<b>1.041</b>	0	0.037	4	<b>1.034</b>	0	0.031	5
	D-D	1	1.219	19	0.189	19	1.125	16	0.115	17	1.071	14	0.065	15	1.063	17	0.058	18	1.058	20	0.053	21
	NB-D	0	1.209	4	0.180	3	1.117	3	0.106	5	1.062	5	0.051	5	1.041	1	<b>0.034</b>	0	<b>1.032</b>	0	<b>0.024</b>	0
	NB-D	1	1.216	14	0.184	4	1.131	21	0.117	18	1.074	16	0.069	17	1.057	14	0.053	15	1.051	13	0.046	14
	I-D	0	1.341	27	0.262	27	1.288	27	0.226	27	1.234	27	0.194	27	1.179	27	0.161	27	1.128	27	0.119	27
	I-D	1	1.344	27	0.262	27	1.294	27	0.228	27	1.237	28	0.194	27	1.188	29	0.161	27	1.140	29	0.118	27
	SMS		<b>1.052</b>	0	<b>0.048</b>	0	<b>1.037</b>	0	<b>0.030</b>	0	<b>1.034</b>	0	<b>0.027</b>	0	<b>1.037</b>	0	<b>0.032</b>	0	<b>1.036</b>	0	0.031	3

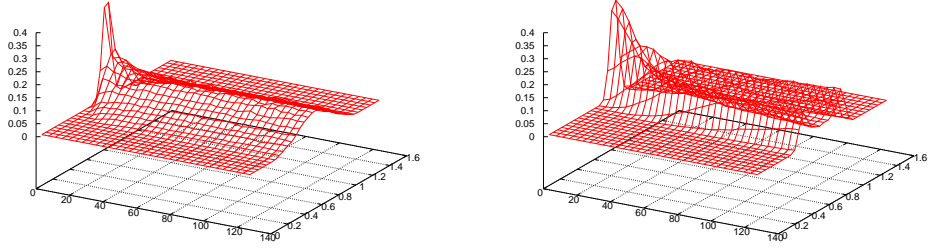
When looking closer at the typical algorithm behavior (Fig. 6 and Fig. 8), one can also observe another obvious difference between the four force-based strategies: The BH-D variant produces consistently more solutions in the middle of the Pareto front while the other strategies, in particular RA-D, are more balanced and outperform the BH-D variant at the extremes (Fig. 8). Overall, we therefore recommend to use BH-D when a focus on the middle of the front is desired and RA-D otherwise because of its simple implementation and the resulting more uniform distribution.

**Distribution of the Population Over the Objective Space.** Several interesting observations can be made about the dynamics of the population in polar coordinates. In Fig. 6, we can see that, for RA-D, solutions are distributed following a bell-shaped distribution, with the mean being the middle of the Pareto front ( $0.8 \simeq \pi/4$ ). However, For I-D, for which the  $\mu$  directions are uniformly distributed over  $[0, \pi/2]$  and constant, the distribution of individuals is seemingly different. In fact, since the Pareto front lies in a smaller range than  $[0, \pi/2]$ , searching along more than one fixed search direction results in reaching the extreme points of the Pareto front, especially for anti-correlated instances. All this is not only observed after the specific generations of Fig. 6 but it can also be seen in Fig. 7 that the point distribution converges rather quickly with time and stays roughly the same after about 60 generations.

**Comparison with SMS.** From Tables 1 and 2, it appears that SMS outperforms the other algorithms for few generations while for a larger number of generations, the force-based approach is competitive with SMS on all instances, and it even outperforms it for the two instances with positive and negative correlation—although the force-based approach uses a simpler selection strategy compared to the global one of SMS. Notice



**Fig. 6.** Distribution of population members in the objective space for the weighted sum version of RA-D (left column), and I-D (right) after 128 generations in polar coordinates ( $x$ -axis: angles from 0 to  $\pi/2$ ), for  $\rho = -0.7$ ,  $S = 0$ ,  $\mu \in \{8, \dots, 512\}$ . The  $y$ -axis refers to the empirical probability of having a solution in the corresponding angle.



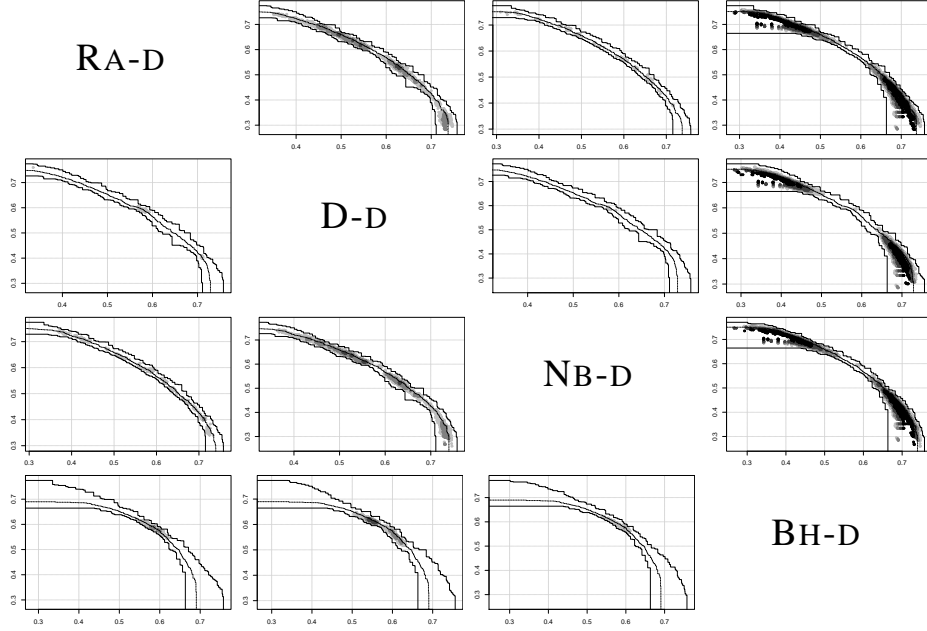
**Fig. 7.** Empirical distribution of population members over generations for RA-D (left) and I-D (right). Runs are for  $\rho = -0.7$ ,  $\mu = 128$ ,  $F = 0$ ,  $S = 0$ .

that the differences between SMS and the proposed approach are rather small for both indicators. When examining the EAF differences between SMS and our approaches, as depicted in the examples of Fig. 9, we observe that SMS produces substantially less solutions at one side of the front while performing substantially better at the rest.

In addition, our comparison between SMS and the force-based approach uses the same number of function evaluations while ignoring the actual costs of maintaining the population and any other complexity issues. In the current implementation, the actual runtimes of the two approaches differ by a factor of about two as SMS needs more costly *global* operations to evolve the population such as sorting and hypervolume computations while in the force-based approach the population replacement can be done *locally* at the level of each solution once the forces are computed.

## 6 Conclusion

In this paper, we proposed a force-based EMO paradigm, and we studied its properties in different configurations. The originality of our approach stems from the fact that each solution computes, dynamically at each generation, a search direction in the ob-



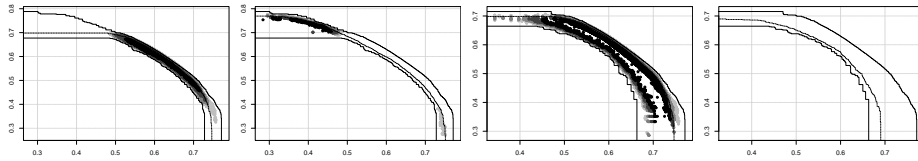
**Fig. 8.** EAF differences for RA-D, D-D, NB-D, and BH-D, with  $\rho = -0.7$ ,  $\mu = 128$ ,  $gen = 128$ ,  $F = 1$  (achievement),  $S = 0$ . The plot in row  $i$  and column  $j$  shows the areas where algorithm  $i$  improves over algorithm  $j$ .

jective space cooperatively, and in parallel with other solutions. Some variants of our approach are proved to be efficient in finding those directions that lead to a good covering of the Pareto front. However, defining what would be the optimal directions that solutions should follow at each step of the search process is a difficult open question. Furthermore, our approach is inherently *local*, in the sense that, although the computation of the directions are “synchronized” for all solutions, the selection is performed locally in parallel for the population members. An interesting open question is to design better direction-based localized strategies, and to come up with inherently distributed approaches. Last but not least, while being in general directly applicable to problems with more than two objectives, further investigations on those problems are needed.

**Acknowledgements.** This work has been supported by the French national research agency (ANR) within the ‘Modèles Numériques’ project NumBBO.

## References

1. Aguirre, H.E., Tanaka, K.: Working principles, behavior, and performance of MOEAs on MNK-landscapes. *Eur J Oper Res* 181(3), 1670–1690 (2007)



**Fig. 9.** EAFs comparing (from left to right): SMS with RA-D ( $F = 0$ ,  $S = 0$ ), RA-D ( $F = 0$ ,  $S = 0$ ) with SMS, SMS with BH-D ( $F = 1$ ,  $S = 0$ ), and BH-D ( $F = 1$ ,  $S = 0$ ) with SMS. Runs are for  $\rho = -0.7$ ,  $gen = 128$ .

2. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. *Eur J Oper Res* 181(3), 1653–1669 (2007)
3. Chen, J.H., Kang, C.W.: A force-driven evolutionary approach for multi-objective 3D differentiated sensor network deployment. *Int. J. of Ad Hoc and Ubi. Comp.* 8(1/2), 85–95 (2011)
4. Hassanzadeh, H.R., Rouhani, M.: A multi-objective gravitational search algorithm. In:  $2^{nd}$  Int. Conf. on Computational Intell., Comm. Sys. and Networks. pp. 7–12 (2010)
5. Hughes, E.J.: Multiple Single Objective Pareto Sampling. In: Congress on Evolutionary Computation (CEC 2003). pp. 2678–2684. IEEE Press (2003)
6. Hughes, E.J.: Many objective optimisation: Direct objective boundary identification. In: Parallel Problem Solving from Nature (PPSN 2008). pp. 733–742 (2008)
7. Kauffman, S.A.: The Origins of Order. Oxford University Press (1993)
8. Khan, J.A., Sait, S.M.: Fast fuzzy force-directed simulated evolution metaheuristic for multiobjective VLSI cell placement. *The Arabian J. for Sc. and Eng.* 32(2B), 264–281 (2007)
9. López-Ibáñez, M., Paquete, L., Stützle, T.: Exploratory analysis of stochastic local search algorithms in biobjective optimization. In: Experimental Methods for the Analysis of Optimization Algorithms, chap. 9, pp. 209–222. Springer (2010)
10. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers (1999)
11. Nobahari, H., Nikusokhan, M., Siarry, P.: Non-dominated sorting gravitational search algorithm. In: Int. Conf. on swarm intelligence (2011)
12. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: A gravitational search algorithm. *Information Sciences* 179(13), 2232 – 2248 (2009)
13. Verel, S., Liefvooghe, A., Jourdan, L., Dhaenens, C.: Analyzing the effect of objective correlation on the efficient set of MNK-landscapes. In: Learning and Intelligent Optimization (LION 5). pp. 116–130. LNCS (2011)
14. Wagner, T., Beume, N., Naujoks, B.: Pareto-, Aggregation-, and Indicator-based Methods in Many-objective Optimization. In: Conference on Evolutionary Multi-Criterion Optimization (EMO 2007). pp. 742–756. Springer (2007)
15. Wang, Y., Zeng, J.C.: A constraint multi-objective artificial physics optimization algorithm. In:  $2^{nd}$  Int. Conf. on Computational Intell. and Natural Computing. pp. 107–112 (2010)
16. Wang, Y., Zeng, J.C., Cui, Z.H., He, X.J.: A novel constraint multi-objective artificial physics optimisation algorithm and its convergence. *Int. J. Innov. Comput. Appl.* 3(2), 61–70 (2011)
17. Zhang, Q., Li, H.: MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation* 11(6), 712–731 (2007)
18. Zitzler, E., Thiele, L., Laumanns, M., Foneseca, C.M., Grunert da Fonseca, V.: Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation* 7(2), 117–132 (2003)